

### **REMARKS**

Reconsideration of the application, as amended, is respectfully requested. With this response, claims 1 and 5 are amended, claims 3 and 7 are canceled, and claims 11-12 are newly added. No new matter is added by amendment. After entering the amendments identified herein, claims 1-2, 4-6, and 8-12 will be pending in the application.

Claims 1, 2, 4-6 and 8-10 were rejected under 35 U.S.C. §103(a) as being unpatentable over Cox (U.S. 6,189,041) in view of Plummer (RFC 826, "An Ethernet Address Resolution Protocol"). Claims 3 and 7 were rejected under 35 U.S.C. §103(a) as unpatentable over Cox in view of Plummer and Aditya (U.S. 5,918,021).

Plummer discloses the well-known, conventional ARP protocol.

Aditya was discussed in a prior reply. In short, it aggregates multiple physical NICs and presents such collection as a virtual NIC having the aggregated bandwidth of the collection. It includes load balancing mechanisms to distribute packets to the underlying physical NICs. Aditya does not discuss load balancing across a cluster of processor nodes.

Cox discloses a next hop resolution protocol (NHRP) that may be used in conjunction with an emulated LAN (ELAN) or the like. For purposes of the present application, its relevant disclosure concerns the disclosed ARP processing. To better understand the distinctions in the amended claims, a review of conventional ARP processing and a review of how this conventional processing is modified in emulated LANs is provided.

Conventional ARP processing allows a first node to determine the MAC (or layer 2) address for another node on a network, if the first node knows the IP (or layer 3) address of the other node. In this way, subsequent to the ARP processing, the first node and second node may communicate with one another via layer 2 addressing (i.e., via a switch) as opposed to requiring layer 3 (and more costly) communication via a router.

The first node issues an ARP request. The ARP request includes the IP and MAC address of the first node (i.e., the ARP requester) and also includes the IP address of the other, target node (i.e., the node for which the requesting node desires the MAC address). The ARP

request is broadcast on the network so the other nodes may observe such. The node associated with the IP address in the ARP request message detects such message. The target node programs its own addressing tables to store the IP-MAC address association of the requesting node. The target node then formulates an ARP reply. The ARP reply, among other things, includes the MAC address of the target node. The ARP reply is unicast (not broadcast) to the ARP requester. The ARP requester receives the ARP reply, and programs its addressing tables to remember the IP-MAC address association of the ARP replier, i.e., the node which the ARP request effectively targeted with the original ARP request. This is known technology, discussed in Plummer and elsewhere.

LAN emulation effectively adds another layer of complexity and processing. When emulating Ethernet functionality on a non-Ethernet underlying network (e.g., a non-broadcast multiple access (NBMA) network) another layer of addressing is involved. An Ethernet MAC address alone is insufficient to instruct an NBMA network where to deliver a packet. NBMA networks need an identification of the point-to-point route from the transmitter to the receiver. Such identification is variously called the “virtual circuit” number, the “virtual channel connection” number (e.g., terminology used in Cox), or the “virtual interface” number (terminology used in the present application)

In LAN emulation contexts, a node will have a conventional stack including an Ethernet driver, but the conventional stack will be supplemented by a lower layer (communication fabric) driver. The Ethernet driver, instead of communicating directly with a link, will communicate with the communication fabric driver, which will effectively emulate the Ethernet link on the underlying network (e.g., NBMA network). Thus, when issuing an ARP request the node will act as described above, but instead of issuing an ARP request directly on a physical Ethernet link, the Ethernet driver will issue the request to a lower layer driver. This lower layer driver communicates with other nodes (e.g., point-to-point) using “virtual circuit” addresses or “virtual interface” addresses; i.e., addresses understood by the underlying network. The lower layer driver will receive the ARP request from the Ethernet driver and add an “extension” to the packet. The extension includes information relevant to the underlying network, including the lower layer fabric address of the requestor. This extended (or “wrapped”) packet is sent to a node that is responsible for emulating the functionality of an Ethernet switch,

and that entity will distribute the ARP request message to all nodes to emulate the broadcast of the ARP request. (This may be accomplished, for example, by sending a point-to-point message to every node in the network.)

The target node detects the message and forms an ARP reply. The lower layer (communication fabric) driver processes the packet extension and programs a table to record the association between the requester's MAC address and virtual interface address (or fabric address), enabling future point-to-point communication from the target node back to the requesting node. It then sends the stripped packet up the stack, which operates as described above. As a result, another layer in the stack will program another table to store the association between the requester's MAC address and IP address. **Thus, two tables are modified: one in the fabric driver to store the association between MAC and virtual interface address, and one in a higher layer to store the association between IP and MAC address.**

The target node forms its ARP reply, which is sent down the stack, and analogously to the request, the lower layer driver will add an extension so that the ARP reply may be unicast back to the original requester, using the virtual interface address of the requester that was remembered when the request arrived. The requester will receive the packet and operate analogously. A lower layer driver will strip out the extension and program a lower layer table to store the association between the target's MAC address and virtual interface address, and send the ARP reply up the stack where a higher layer will program a table to store the association between the target's IP and MAC addresses.

Cox discloses the above and further adds the cut-through feature noted by the examiner. With the cut through feature, subsequent unicast messages between the ARP replier and the ARP requester need not involve (or go through) the node emulating the Ethernet functionality.

The amended claims have two principal distinctions over the cited art: the first focuses on the specific context of load balancing clusters; the second concerns a significant improvement in the processing and programming of the address associations.

Amended claims 1 and 5 now recite a method and system in which a subset of the processors are organized as a cluster and in which one of the processors in the cluster is a load balancing processor node. When any processor in the cluster issues an ARP request, the ARP request is programmed to include a MAC address for the load balancing processor node. (As a consequence of ARP semantics, this will cause the ARP reply to be unicast to the load balancing node, not the ARP requester.) The claims further recite that when the load balancing node receives an ARP reply, it distributes said reply to all nodes in the cluster. (Thus all of the nodes will then be able to program the address association of the ARP replier.) Support for these claims is found throughout the sections on ARP processing including page 21 of the specification.

Cox and Plummer are silent on load balancing, as noted by the examiner. Aditya balances messages among physical NICs, and is not concerned with ARP processing or any form of modifying ARP requests as specified in the amended claims. Consequently claims 1-2, 4-6, and 8-10 should be found allowable.

Newly added claims 11 and 12 are directed to a significant improvement in the handling of lower layer addresses in emulated networks. Specifically, the underlying fabric addresses are expressed in a MAC address format with predefined setting of select address bits to identify the MAC address as carrying virtual interface address information within the MAC address format. In this manner, the communication fabric driver need not keep a table to map a fabric address with a MAC address (as described above) and instead the conventional ARP table is used to store a direct translation from an IP address to a fabric address (which happens to be expressed in MAC address format). Only one table look-up is needed for communication, as opposed to two. This improvement is significant when one considers the look-up cost typically associated with the sparsely populated tables inherent with 48-bit MAC addresses. None of the art discloses or suggests such features. Support is found throughout the sections concerning network addressing, including page 9 and 11.

Applicant believes that this feature was captured (or at least that was the intent) in the original claims by their reference to programming an ARP table (i.e., singular) in contrast to the

conventional LAN emulation which requires two tables to be programmed. The new claims hopefully make the distinction clearer.

In view of the above amendment, Applicant believes the pending application is in condition for allowance. A Petition for a three-month extension of time accompanies this reply, and the Commissioner is hereby authorized to charge Deposit Account No. 08-0219 the fee of \$510 to cover the cost of this extension. Please also charge the \$395 fee for the Request for Continued Examination, which also accompanies this reply, to our Deposit Account No. 08-0219. No other fees are believed to be due at this time; however, please apply any charges not covered, or any credits, to Deposit Account No. 08-0219.

Dated: April 6, 2006

Respectfully submitted,

By 

Peter M. Dichiara

Registration No.: 38,005

WILMER CUTLER PICKERING HALE AND  
DORR LLP

60 State Street

Boston, Massachusetts 02109

(617) 526-6000

Attorneys for Applicant